Aufgabe: Eine LED soll pulsierend heller und dunkler werden. (Auch als engl. „faden“ bezeichnet)

Material:

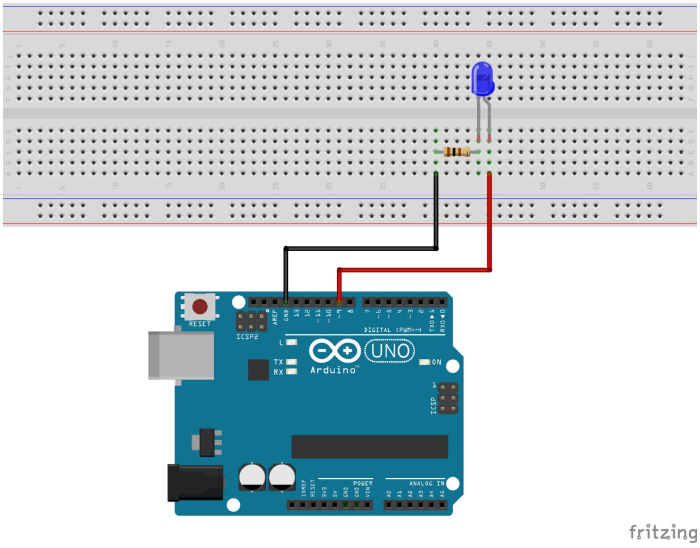
- Arduino-Board
- Eine blaue LED
- Ein Widerstände mit 100 Ohm
- Breadboard
- Steckkabel

In dieser Anleitung lernst du:

- Pulsweitenmodulation verwenden
- Mathematische Funktionen im Code
- Vergleich: „Oder“
- Eine LED anschließen
- Einen Widerstand vorschalten
- IF-Befehl

Der Arduino ist ein digitaler Mikrocontroller. Er kennt an seinen Ausgängen nur „5 Volt an“ oder „5V aus“. Um die Helligkeit einer LED zu variieren, müsste man die Spannung jedoch variieren können. Zum Beispiel 5V wenn die LED hell leuchtet und 4 Volt, wenn sie etwas dunkler leuchtet usw. Das funktioniert an digitalen Pins nicht. Es gibt jedoch eine Alternative. Sie nennt sich Pulsweitenmodulation (PWM). Die PWM lässt die 5V Spannung pulsieren. Die Spannung wird also im Millisekundenbereich ein und ausgeschaltet. Bei einer hohen PWM liegt das 5V Signal nahezu durchgehend am jeweiligen Pin an. Bei einer geringen PWM ist das 5V Signal kaum noch vorhanden (Da dies eine sehr kompakte Zusammenfassung ist, sollte man sich ggf. detaillierte Quellen zum Themengebiet PWM ansehen). Mit dieser PWM kann man bei LEDs einen ähnlichen Effekt erreichen, als würde man die Spannung variieren. Nicht alle digitalen Pins am Board haben die PWM Funktion. Die Pins an denen die PWM funktioniert sind besonders gekennzeichnet, bspw. durch eine kleine Welle vor der Zahl mit der Pinnummer .

Aufbau:



Der folgende Sketch erzeugt ein pulsierendes Licht an einer Leuchtdiode. Die Beschreibung ist in diesem Fall etwas umfangreicher. Für eine bessere Übersichtlichkeit wird der Sketch zunächst ohne Beschreibungen dargestellt und im Anschluss erneut mit den entsprechenden Erklärungen.

```
int LED=9;
int helligkeit=0;
int fadeschritte=5;

void setup()
{
pinMode(LED, OUTPUT);
}

void loop()
{
analogWrite(LED, helligkeit);
helligkeit=helligkeit + fadeschritte;
delay(25);
if(helligkeit==0 || helligkeit== 255)
{
fadeschritte= -fadeschritte;
}
}
```

Dies ist der Sketch mit Erklärungen:

```
int LED=9;              //Das Wort „LED“ steht jetzt für den Wert 9.
int helligkeit=0;       //Das Wort „helligkeit“ steht nun für den Wert, der bei
                        derPWM ausgegeben wird. Die Zahl 0 ist dabei nur
                        einbeliebiger Startwert.
int fadeschritte=5;     //fadeschritte: bestimmt die Geschwindigkeit des
                        „fadens“

void setup()
{                       //Hier beginnt das Setup.
pinMode(LED, OUTPUT); //Der Pin mit der LED (Pin9) ist ein Ausgang
}

void loop()
{
analogWrite(LED, helligkeit);
//Mit der Funktion analogWrite wird hier an dem Pin mit der LED (Pin9) die PWM
Ausgabe aktiviert. Der PWM-Wert ist der Wert, der unter dem Namen „helligkeit“
gespeichert ist. In diesem Fall „0“ (Siehe ersten Programmabschnitt)
helligkeit=helligkeit + fadeschritte;
//Nun wird der Wert für die PWM-Ausgabe verändert. Unter dem Wert „helligkeit“
wird jetzt zur vorherigen helligkeit der Wert für die fadeschritte addiert. In
diesem Fall: helligkeit=0+ 5. Der neue Wert für „helligkeit“ ist also nicht mehr
0 sondern 5. Sobald der Loop-Teil einmal durchgelaufen ist, wiederholt er sich.
Dann beträgt der Wert für die Helligkeit 10. Im nächsten Durchlauf 15 usw. usw…
delay(25);
//Die LED soll für 25ms (Millisekunden), also nur ganz kurz die  Helligkeit
beibehalten. Verringert man diesen Wert, wird das Pulsieren ebenfalls schneller.
if(helligkeit==0 || helligkeit== 255){
//Bedeutung des Befehls: Wenn die Helligkeit den Wert 0 ODER 255 erreicht hat,
wechselt der Wert für die „fadeschritte“ von positiv zu negativ bzw. andersrum.
Grund: Die LED wird zunächst bei jedem Durchlauf des Loop-Teils immer ein
bisschen heller. Allerdings ist irgendwann der Maximalwert für die PWM-Ausgabe
mit dem Wert 255 erreicht. Die LED soll dann wieder Schritt für Schritt dunkler
werden. Also wird der Wert für die „fadeschritte“ an dieser Stelle negativiert
(Ein Minuszeichen wird davor gesetzt)
fadeschritte= -fadeschritte;
//Das bedeutet für den nächsten Durchlauf, dass in der Zeile „helligkeit =
helligkeit + fadeschritte;“ die helligkeit abnimmt. Beispiel: „helligkeit=255+(-
5)“. Der Wert für Helligkeit ist ab dann 250. Im nächsten Durchlauf 245 usw.
usw… Sobald der Wert für Helligkeit bei 0 angekommen ist, wechselt wieder das
Vorzeichen. (Man bedenke die alte mathematische Regel: „minus und minus ergibt
plus“.)
}// Mit dieser letzten Klammer wird der IF-Befehl geschlossen.
}// Mit dieser letzten Klammer wird der Loop-Teil geschlossen.
```